

//SNAKE

OBJECTIVES

- Group project
- Practice
 - Coding
 - Source control
- Learn
 - basic Game Engine Architecture
 - Group work
- Expected
 - Playable game

- This is a Graded project
- Submission: 13 October 23h55

1

// THE GAME

GAMEPLAY 1/2

- Setup:
 - Player controls the Snake Direction
 - The snake is always moving forward
 - The speed of the snake is increasing every ?* seconds
 - Limited rectangular Space: Arena
 - Visual Boundaries
 - Food is spawned periodically randomly in the arena every ?* seconds
 - If the snake eats food, his body grows ?* additional units
 - There can be a maximum of ?* food in the arena

GAMEPLAY 2/2

- Points:
 - Every seconds the snake is alive, he gains ?* points
 - Every time the snake eats food, he gains ?* points
- Game Over
 - Snake head hits a wall
 - Snake head eats Snake Body



DIVIDE AND CONQUER

Break down in steps:

1. Setup the project
2. Move the Snake
3. Setup an Arena with boundaries
4. Kill the Snake if he touches the boundaries
5. Add food to give points
6. Food makes the snake longer
7. Kill the snake when he eats the snake
8. Add a Menu
9. Etc.

2

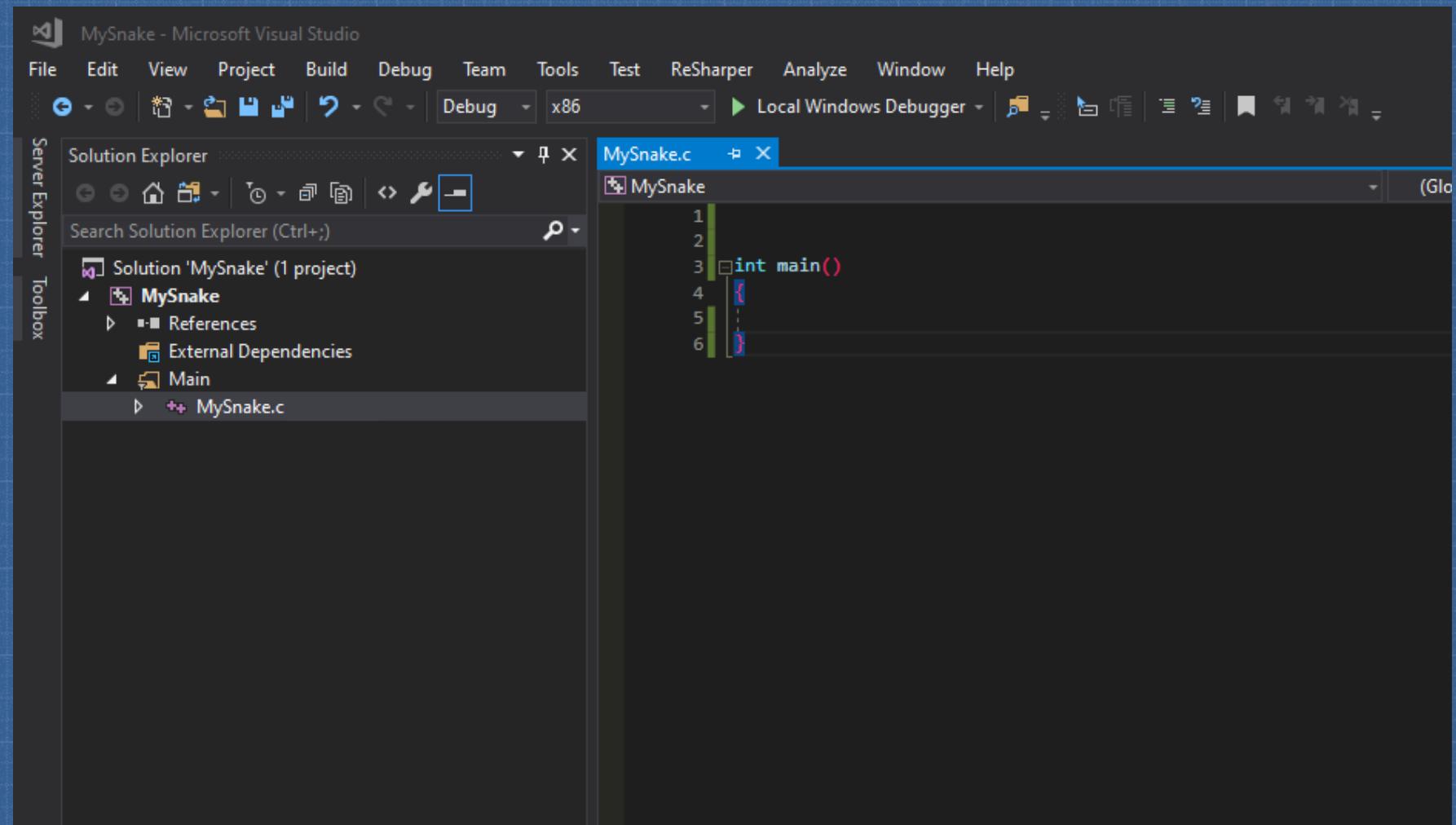
// LET' S START

Project Setup Live Demo

SETUP PROJECT: LIVE DEMO

- Create a new Solution
 - Wizard: Windows Console Application
 - Remove Precompiled header
 - Organize the Filters/Files
 - Rename Main file from .cpp to .c
 - Remove everything beside the Main function

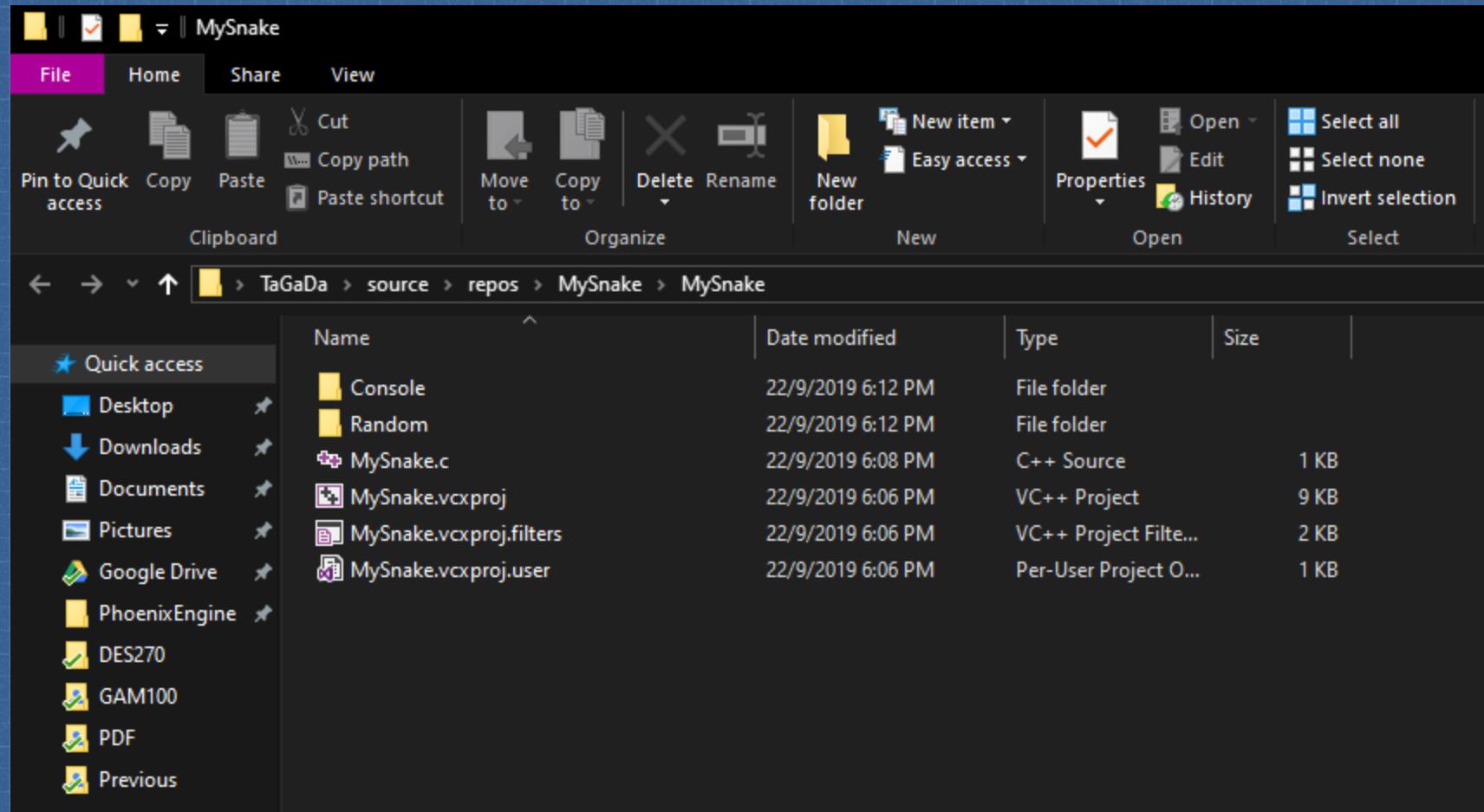
SETUP PROJECT



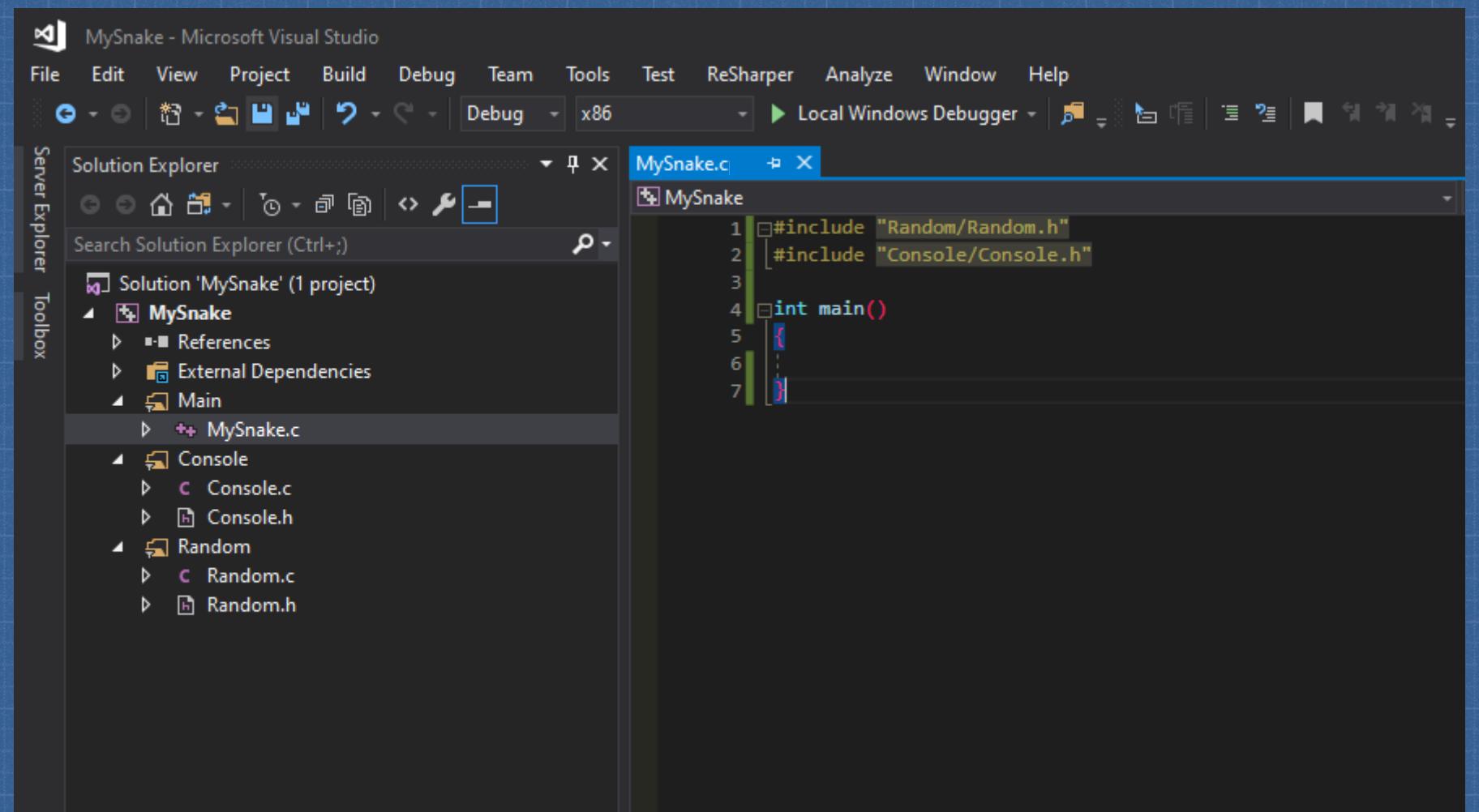
SETUP PROJECT

- Add the provided Libraries to your folder on your HDD
 - Console
 - Random
- Add the “Existing File” in your visual studio solution

SETUP PROJECT: ADD LIBRARIES ON THE DISK



SETUP PROJECT: ADD LIBRARIES TO THE SOLUTION



SETUP PROJECT: CODE INITIALIZATION

```
#include "Random/Random.h"
#include "Console/Console.h"
#include <Windows.h>

static int bGameIsRunning = 1;

int main()
{
    // Game Initialization
    Console_Init();
    Random_Init();

    // Game Loop
    while (bGameIsRunning)
    {
        Sleep(10);
    }

    // Game Shutdown
    Console_CleanUp();
}
```

- We start by setting up a game Loop.
- Here we have a never-ending loop, your program will never stop

SETUP PROJECT: DEFINE WINDOW PROPERTIES

```
#include "Random/Random.h"
#include "Console/Console.h"
#include <Windows.h>

static int bGameIsRunning = 1;

int main()
{
    // Game Initialization
    Console_Init();
    Random_Init();

    // Set a Title to my window
    Console_SetTitle("Snake Test");

    // Setup a Square font and the window size
    Console_SetSquareFont();
    Console_SetWindowedMode(40, 40, false);

    // Display something
    Console_Printf(20, 20, "Hello DigiPen");
    Console_Putchar(20, 21, 'o');

    // Game Loop
    while (bGameIsRunning)
    {
        Sleep(10);
    }

    // Game Shutdown
    Console_CleanUp();
}
```

- we add some code to properly initialize the program
- We also test the Console library output (printf/putchar)

SETUP PROJECT: READ INPUTS

```
#include "Random/Random.h"
#include "Console/Console.h"
#include <Windows.h>

static int bGameIsRunning = 1;

int main()
{
    // Game Initialization
    Console_Init();
    Random_Init();

    // Set a Title to my window
    Console_SetTitle("Snake Test");

    // Setup a Square font and the window size, and
    // remove the cursor visibility
    Console_SetSquareFont();
    Console_SetWindowedMode(40, 40, false);
    Console_SetCursorVisibility(0);

    int x = 0;
    int y = 0;
```

```
// Game Loop
while (bGameIsRunning)
{
    // Move the Player
    if (GetAsyncKeyState(VK_RIGHT))
        x += 1;

    if (GetAsyncKeyState(VK_LEFT))
        x -= 1;

    if (GetAsyncKeyState(VK_ESCAPE))
        bGameIsRunning = 0;

    // Clear the Display, and draw the Player
    Console_Clear();
    Console_Putchar(x, y, 'o');

    Sleep(10);
}

// Game Shutdown
Console_CleanUp();
```

SETUP PROJECT: READ INPUTS

- We test the input,

```
#include "Random.h"
#include "Console/Console.h"
#include "Windows.h"

static bool bGameIsRunning = true;
int x = 0;
int y = 0;

int main()
{
    // Game Initialization
    Console_Init();
    Random_Init();

    // Set a Title to my window
    Console_SetTitle("Snake Test");

    // Setup a Square font and the window size, and
    // remove the cursor visibility
    Console_SetSquareFont();
    Console_SetWindowedMode(40, 40, false);
    Console_SetCursorVisibility(0);

    // We move a 'o' character left or
    // right depending on the input
    int y = 0;
}
```

- There are other methods to

check inputs (windows event system)

- We move a 'o' character left or right depending on the input

```
// Game Loop
while (bGameIsRunning)
{
    // Move the Player
    if (GetAsyncKeyState(VK_RIGHT))
        x += 1;

    if (GetAsyncKeyState(VK_LEFT))
        x -= 1;

    if (GetAsyncKeyState(VK_ESCAPE))
        bGameIsRunning = 0;

    // Clear the Display, and draw the Player
    Console_Clear();
    Console_Putchar(x, y, 'o');

    Sleep(10);
}

// Game Shutdown
Console_CleanUp();
```

SETUP PROJECT: CLEAN UP

```
#include "Random/Random.h"
#include "Console/Console.h"
#include <Windows.h>

// Move these to a global file
static int bGameIsRunning = 1;
static int x = 0;
static int y = 0;

// Move this to another file
void GameInit()
{
    // Game Initialization
    Console_Init();
    Random_Init();

    // Set a Title to my window
    Console_SetTitle("Snake Test");

    // Setup a Square font and the window size, and
    // remove the cursor visibility
    Console_SetSquareFont();
    Console_SetWindowedMode(40, 40, false);
    Console_SetCursorVisibility(0);
}
```

1/3

```
// Move this to another file
void ReadInput()
{
    // Move the Player
    if (GetAsyncKeyState(VK_RIGHT))
        x += 1;

    if (GetAsyncKeyState(VK_LEFT))
        x -= 1;

    if (GetAsyncKeyState(VK_ESCAPE))
        bGameIsRunning = 0;
}

// Move this to another file
void RenderScene()
{
    // Clear the Display, and draw the Player
    Console_Clear();
    Console_Putchar(x, y, 'o');
}

// Move this to another file
void GameLoop()
{
    ReadInput();
    RenderScene();
}
```

2/3

SETUP PROJECT: CLEAN UP

```
// Move this to another file
void GameShutdown()
{
    Console_CleanUp();
}

// Simplified Program entry point
int main()
{
    GameInit();

    while (bGameIsRunning)
    {
        GameLoop();
        Sleep(10);
    }

    GameShutdown();
}
```

3/3

- We clean up by separating the code into functions to:
 - Help make the code easier to read
 - Help multiple people to work on them
- We should move these functions into separate files ie :
Engine.c/Engine.h,
Game.c/Game.h ... etc.

YOUR TURN

- Next steps
 - Make the snake move up / down
 - Make the snake move automatically
 - Make the snake die when he moves out of the window
 - Etc.